(2)

NCS TIB 90-16

NATIONAL COMMUNICATIONS SYSTEM

TECHNICAL INFORMATION BULLETIN 90-16

COMPARATIVE ANALYSIS OF GRAY SCALE CODING TECHNIQUES FOR GROUP 4 FACSIMILE

AUGUST 1991

OFFICE OF THE MANAGER
NATIONAL COMMUNICATIONS SYSTEM

WASHINGTON, D.C. 20305

August 1991 Final

Comparative Analysis of Gray Scale Coding Techniques for Group 4 Facsimile

C-DCA100-87-C-0078

Delta Information Systems, Inc.
Horsham Business Center, Bldg. 3 Suite 120
300 Welsh Road
Horsham, PA 19044-2273

National Communications System
Office of Technology & Standards
701 S. Court House Road
Arlington, VA 22204-2199

NCS TIB 90-16

Approved for public release; distribution is unlimited.

This compares bit-plane coding (BPC), differential pulse code modulation (DPCM), transform coding, and predictive vector quantization (PVQ) to determine their relative effectiveness for the addition of gray scale to Group 4 facsimile. At present, the CCITT Recommendation for Group 4 facsimile permit the transmission of only black and white imagery (black print on white paper; not photographic). Consequently, most gray scale imagery (like photographs) are usually distorted by Group 4 equipments. To correct this, the CCITT is planning to add gray scale to the Group 4 Recommendations as an option. The aforementioned coding techniques have already been studied individually, but they haven't been compared to one another, nor have they been ranked according to their suitability for Group 4 facsimile.

Gray Scale Coding
Group 4 Facsimile

52

Unclassified Unclassified Unclassified Unlimited

NCS TECHNICAL INFORMATION BULLETIN 90-16

COMPARATIVE ANALYSIS OF GRAY SCALE CODING TECHNIQUES

AUGUST 1991

PROJECT OFFICER

Stephen Perschau

STEPHEN PERSCHAU
Computer Scientist
Office of NCS Technology
and Standards

APPROVED FOR PUBLICATION:

Dennis Bodson

DENNIS BODSON
Assistant Manager
Office of NCS Technology
and Standards

FOREWORD

Among the responsibilities assigned to the Office of the Manager, National Communications System, is the management of the Federal Telecommunication Standards Program. Under this program, the NCS, with the assistance of the Federal Telecommunication Standards Committee identified, develops, and coordinates proposed Federal Standards which either contribute to the interoperability of functionally similar Federal telecommunication systems or to the achievement of a compatible and efficient interface between computer and telecommunication systems. In developing and coordinating these standards, a considerable amount of effort is expended in initiating and pursuing joint standards development efforts with appropriate technical committees of the International Organization for Standardization, and the International Telegraph and Telephone Consultative Committee of the International Telecommunication Union. This Technical Information Bulletin presents and overview of an effort which is contributing to the development of compatible Federal, national, and international standards in the area of facsimile. It has been prepared to inform interested Federal activities of the progress of these efforts. Any comments, inputs or statements of requirements which could assist in the advancement of this work are welcome and should be addressed to:

Office of the Manager
National Communications System
ATTN: NCS-TS
Washington, DC 20305-2010

COMPARATIVE ANALYSIS OF GRAY SCALE CODING TECHNIQUES FOR GROUP 4 FACSIMILE

August, 1991

SUBMITTED TO:
NATIONAL COMMUNICATIONS SYSTEM
Office of Technology and Standards
WASHINGTON, D.C. 20305

Contracting Agency:
DEFENSE COMMUNICATIONS AGENCY
Contract Number - DCA100-87-C-0078
Task Number - 88-002

DELTA INFORMATION SYSTEMS, INC
Horsham Business Center, Bldg. 3, Ste 120
300 Welsh Road
Horsham, PA 19044-2273
TEL: (215) 657-5270 FAX: (215) 657-5273

CONTENTS

ILLUSTRATIONS

TABLES

1.0 INTRODUCTION

This document summarizes work performed by Delta Information Systems, Inc., for the Office of Technology and Standards of the National Communications System, an organization of the U. S. Government, under Task 002 of contract number DCA100-87-C-0078. The purpose of this Task is to compare bit-plane coding (BPC), differential pulse code modulation (DPCM), transform coding, and predictive vector quantization (PVQ) to determine their relative effectiveness for the addition of gray scale to Group 4 facsimile.

At present, the CCITT Recommendations for Group 4 facsimile permit the transmission of only black and white imagery (black print on white paper; not photographic). Consequently, most gray scale imagery (like photographs) are usually distorted by Group 4 equipments. To correct this, the CCITT is planning to add gray scale to the Group 4 Recommendations as an option. The aforementioned coding techniques have already been studied individually, but they haven't been compared to one another, nor have they been ranked according to their suitability for Group 4 facsimile.

Section 2.0, "*Selected Gray Scale Data Compression Techniques*," discusses the four compression techniques whose output images will be compared.

Section 3.0, "*Image Selection*," discusses which image originals were used and why.

Section 4.0, "*Subjective Measurement of Coder Performance*," discusses how the subjective measurements were designed to elicit the performance of the data compression techniques, as compared to one another, and with regards to image quality and compression.

Section 5.0, "*Results and Analysis of Subjective Comparison*," presents the results of the subjective comparison and analyzes them.

Section 6.0, "*Recommendations*," proposes which compression technique(s) should be used by Group 4 for transmitting gray scale imagery.

2.0 SELECTED GRAY SCALE DATA COMPRESSION TECHNIQUES

Data compression techniques are capable of both lossless and lossy compression. Lossless (or reversible, or noiseless coding, or redundancy reduction) compression preserves the original image; facsimiles made using these methods are exact duplicates of the originals. On the other hand, lossy (or irreversible, or fidelity-reducing coding, or entropy reduction) compression makes no attempt to preserve the original; facsimiles made using these methods are approximations of the original.

Another major difference between the two is that with lossless compression techniques, the compression achieved is dependent upon the amount of redundancy found in the original while with lossy compression techniques a given compression may be achieved by discarding more and more information (with subsequently poorer and poorer renditions of the original).

The four data compression techniques selected for comparison are all lossy data compression techniques with a few of them being capable of lossless compression. In addition, more than one of these techniques rely on the arithmetic coder.

2.0.1 Arithmetic Coder

Consider the case of a bi-level image (black and white, no intermediate gray levels). Uncompressed data for such an image requires 1 bit per pixel. As the binary image is scanned, runs of two or more pixels of the same value are very frequently encountered, and long runs of the same value are common. The Arithmetic Coder[1],[2] continuously keeps track of the local probabilities (frequencies averaged over many pixels, but not the whole image) of the two symbols ("black" and "white"). The one currently occurring the more frequently is called the more probable symbol (MPS), and the other is called the less probable symbol (LPS). The more probable the MPS, the lower the bit rate for it, and the higher the bit rate for the LPS. Since the MPS occurs more frequently, the average bit rate for both symbols decreases as the probability of the MPS increases.

The Arithmetic Coder adapts to local statistics. If, for example, a long run of "blacks" is followed by a long run of "whites," high compression is achieved during most of both runs, but the bit rate per symbol increases considerably during the transition between them. Because of these transitions, random binary data, in general, give no compression, and sometimes give expansion (more than one bit per original bit).

What was just described is an example of a single-context model. Most systems employing the Arithmetic Coder require a multiple-context model. For example, to code multiple-way decisions, the decisions must be mapped into binary trees, the Arithmetic Coder encoding each binary decision in a given tree. Each decision in the tree may have different statistics from those of other decisions in the same tree, and should therefore be considered in a separate context to take advantage of these different statistics. Another example of multiple contexts is a set of binary decisions, not necessarily comprising one tree, having significantly different statistics.

The Arithmetic Coder can track separately and simultaneously any reasonable number of contexts, limited only by available memory, keeping local statistics for each. The contexts share a common probability table; hence, each context requires storage only for its MPS value and a pointer into the probability table. Each context exhibits high compression when its MPS is much more frequent than its LPS. Its bit rate increases only when the frequency of its LPS increases.

Compression can be further enhanced by taking advantage of correlation among various binary decisions. This is called conditioning, and is employed widely in the JPEG Arithmetic Coder models. For example, binary image compression could be improved by the use of two contexts: (1) current pixel is preceded by a white pixel, and (2) current pixel is preceded by a black pixel. The current pixel would most of the time be white in context 1 and black in context 2.

2.1 Bit Plane Coding (BPC)

Bit plane coding requires the storage of the entire image into memory prior to transmission. The light intensity of each pixel of the input image is digitally represented by an n-bit value (usually 8). This n-bit per pixel image is then split into n-bit planes with each plane

representing one of the bits in the original pixel. Bitplane coding encodes the most significant bits (MSB) of each pixel first, successively followed by each less significant bit plane.

The most significant bit plane is encoded by either a conventional black-white encoder such as the modified Read code or an implementation of the arithmetic coder. After the transmission of the MSB, the encoder transmits the next significant bitplane, corresponding to the next significant bit position (n-1) of the input pixel. This process continues until all of the bit planes are transmitted. At the receiver end, each bitplane is reconstructed plane by plane starting with the (most/least) significant plane and decoding until all planes are processed.

In a practical implementation of bit plane coding the image is stored in Pulse Code Modulation (PCM) form (See Section 2.2) in memory on a line by line basis prior to transmission. Each line is then transmitted one at a time by the bit plane process. By transmitting and receiving the entire n-bitplanes corresponding to each individual bit in every m-bit image pixel, the output image is an exact reproduction of the input image. Nevertheless, in practice, bit plane coding compression techniques require a quantization step prior to the encoding process in order to achieve bit rates comparable to those of other gray scale compression techniques. Starting with the least significant bitplane each m-bit pixel is truncated to an m-1,m-2,m-3,..etc. bit pixel. *The remaining n-bit pixels are then coding into their* respective bitplanes (See equation 1). As more bitplanes are removed from the original image, compression is improved at the expense of reduced image quality.

Equation 1:

$$(m - q) = n \text{ remaining pixel bits}$$

Where m = original input pixel bits
q = number of removed bits

A BPC process is comprised of five process steps: quantization, bitplane separation, encoding, decoding, and reconstruction.

2.1.1 Quantization

The algorithm begins with an n-bit quantization of the input image pixels with gray code substitution; where n represents the number of bits used during transmission.

During the quantization step, each m-bit pixel in the input image is converted to an n-bit pixel by discarding the (m-n) least significant bits of original pixel. This quantization process approximates the input image pixel values with n bits of precision. For improved compression the n-bit binary value is replaced with the corresponding gray code depending on the gray code value.

By substituting the Gray codes for the binary pixel values, the number of bits toggling between 0 and 1 is reduced. The gray codes are designed so that only one bit changes state between adjacent intensity levels. For example, 7 = 1010 and 8 = 1011 in the 4-bit case. In contrast, one or even all of the bits of the uncoded binary value can change state between levels. Without graycodes 7 is represented by 0111 and 8 is represented by 1000. In this example all of the bits change state. If a gray area in a 4-bit image continually toggles between levels 7 and 8, all for bit planes would contain state changes for every bit when uncoded binary values are used. Nevertheless, only one bit plane would change state when the Gray codes are used. Without gray codes many more short bit runs for the encoder and decoder occur. With gray coding employed bit run lengths of each bitplane are increased.

When quantization is employed the BPC compression technique does not produce output images that are identical to the input images. There is some image quality degradation that increases as the quantization becomes more severe. Nevertheless, if the n-bit quantization step is omitted, BPC would transmit the m-bit gray scale images with zero distortion. To achieve compression levels comparable to other techniques, quantization must be used to some degree.

2.1.2 Bitplane Separation

The second step is the separation of n input pixel bits into their respective bitplanes.

During bit plane separation, the n-bit Gray coded image is divided into n separate binary images by using one bit from each n-bit pixel. All bits within a given bitplane originate from one and only one unique bit position in the input pixel. Thus bit planes are ranked from most significant to least significant just as their respective bits are ranked in the input pixel. Of course the number of bit planes for encoding is a 1 to 1 relationship with the number of bits in the input pixel.

2.1.3 Coding and Decoding

One-dimensional or two-dimensional binary encoding concludes the transmission process where the bitplanes are compressed as individual binary images. A number of coders can be used to encode the bit planes, such as Modified Read Coding (MRC) or arithmetic coding. For this study, we used arithmetic coding.

When decoded, each plane is exactly reconstructed by the appropriate MRC, or arithmetic decoder algorithm.

2.1.4 Reconstruction

Reception begins with one- or two-dimensional binary decoding followed by an n-bit reconstruction of the input image.

During n-bit reconstruction, each bitplane is employed according to the input pixel's bit position that the bitplane represents. Each n-bit pixel is reconstructed by extracting one pixel from each of the respective bit planes. The reconstructed pixel is now in Gray coded form which must be decoded into the appropriate n-bit binary value. The most significant bits of the binary code are placed in the most significant bit positions of the m-bit pixel. The image is output as an m-bit image so that the printing method at the point of reception is identical to that at the point of transmission.

2.2 Differential Pulse Code Modulation (DPCM)

Pulse Code Modulation (PCM) transmits each pel as an independent sample without taking advantage of the high degree of pel-to-pel correlation existing in most pictures. *Differential* Pulse Code Modulation (DPCM) takes advantage of the pel-to-pel correlation by using the local statistical information when encoding gray scale images.

The difference between a prediction of the gray level of an input pixel and its actual value is used to achieve compression. The predictor, which is implemented in both the transmitter and receiver, estimates the brightness of the present input pixel based upon information previously transmitted about preceding pixels. At the transmitter, the predicted signal is subtracted from the input and this difference is quantized, digitized to n (usually 3) bits and transmitted. At the receiver, the error value is added to the predicted value, producing a close approximation to the original input gray level value. The requirement of using an identical predictor in both the transmitter and receiver becomes readily apparent.

DPCM compression is often enhanced by performing additional steps, and usually consists of six processes: spatial filtering (optional), subsampling (optional), interpolation (reverses subsampling on receiver side), prediction, quantization, and coding. For this report, three levels of compression were obtained using a 3-bit DPCM in conjunction with spatial filtering and subsampling. For the lowest level of compression, the 3-bit DPCM was used without spatial filtering or subsampling. For a medium level of compression, the 3-bit DPCM was used with a one-dimensional spatial filter, followed by subsampling. For the highest level of compression, the 3-bit DPCM was used with a two-dimensional spatial filter, followed by subsampling.

2.2.1 Spatial Filtering

Spatial filtering, or smoothing, was applied to reduce noise in the image and to increase compression. It is an optional pre-processing step consisting of applying a horizontal spatial filter to the input pixels.

For the one-dimensional filter, the gray level value of each pixel is filtered by applying the following equation:

$$I_o' = I_o + \sum_{m=-N}^{N} A_m \Delta_m$$

where I_o' is the filtered value

I_o is the actual gray level value of the pixel

Δ_m is the difference between the m^{th} pixel and the present pixel, $I_m - I_o$

A_m is the weighting coefficient associated with the m^{th} pixel

N is the number of pixels away from the present pixel

On the first pass, $N = 1$ and $(A_{-1}, A_1) = (¼, ¼)$, On the second pass $N = 2$ and $(A_{-2}, A_{-1}, A_1, A_2) = (⅛, ¼, ¼, ⅛)$. To limit the effect of the filter on the image, the difference, Δ_m, is restricted to a maximum of about 6 percent of the gray level range (16 for an 8-bit image) on the first pass and about 1.5 percent (4 for an 8-bit image) on the second pass.

To further reduce noise and increase compression, a *two-dimensional* spatial filter can be used. It is similar to the one-dimensional filter, except it takes advantage of correlation between adjacent lines. Values which might be used for the filter coefficients are shown in Figure 2-1.

1	4	1
4	16	4
1	4	1

Figure 2-1. Two-Dimensional Filter Coefficients

X	O	X^C	O	X^F
O	X^B	O^A	X^D	O
X	O	X^E	O	X
O	X	O	X	O
X	O	X	O	X

X - Transmitted Pixel
O - Untransmitted Pixel

Figure 2-2. Horizontal Staggered Subsampling Pattern

2.2.2 Subsampling and Interpolation

Subsampling was done in a staggered, or checkerboard, pattern at a 2:1 rate (See Figure 2-2). This pattern was chosen for it's ability to allow for an efficient interpolation method at the receiver. The interpolation of each untransmitted pixel is achieved by taking the average of the four surrounding transmitted pixels. For example, Figure 2-2 shows the gray level value of the untransmitted pixel A is approximated by A = (B+C+D+E)/4. The subsampling pattern and corresponding interpolation method are designed to minimize the visual effect of the loss of resolution caused by the 2:1 subsampling.

2.2.3 Predictor

The predictor used in the DPCM compression technique is very important, and a well designed predictor is vital to achieving good compression while maintaining image quality. Predictors can vary greatly in complexity. For example, a predictor can estimate solely on the basis of the preceding pixel, or for a more accurate prediction, it can use several preceding pixels. A typical predictor which exhibits good compression while maintaining good image quality is a three-neighbor gray level value predictor.

Either preprocessed pixel values or unprocessed pixel values can provide the input to a DPCM compression algorithm. For example, the 3-bit DPCM reduces the bit rate for transmission from 8 bits/pixel to 3 bits/pixel. The prediction equation used in determining the difference image depends on whether the input pixels were subject to preprocessing or not. For the 3-bit DPCM, the following two equations are used in predicting the difference image:

$$D = 0.5A + 0.25(B + C), \text{ without subsampling, or}$$

$$D = 0.3B + 0.35(C + F), \text{ with subsampling}$$

where A,B,C, and F are the reconstructed gray level values of previously transmitted pixels, as illustrated in Figure 2-3. Since the actual gray level values are not available for the prediction at the receiver, the reconstructed neighbor gray level values are used in the prediction equation instead of the actual values.

$$\begin{array}{ccccc} X & X & X^B & X^C & X^E \\ X & X & X^A & X^D & X \end{array}$$

(a) Unsubsampled Image

$$\begin{array}{ccccc} X & O & X^C & O & X^F \\ O & X^B & O & X^D & O \end{array}$$

(b) Subsampled Image

X - Transmitted Pixel
O - Untransmitted Pixel

Figure 2-3. Pixels Used in Prediction Equations

2.2.4 Quantizer

Once a predicted value for a pixel is known, the difference or error value between the predicted value and the actual value can be calculated and quantized to n-bits (3 in our case). Quantizers often vary from one bit to five bits per pixel depending upon their use. Nevertheless, non-linear three bit quantizers are often

employed, and are more precise when encoding small error values, but are less precise when encoding large error values.

2.2.5 Coding

After the error value has been quantized, the number corresponding to the quantizer level is transmitted, and entropy coders (like Huffman coders) are typically employed to further compress the output. Results achieved with entropy coding applied in this fashion are typically about one bit/pixel.

The Huffman variable-length encoding further reduces the transmitted number of bits per pixel by exploiting the local conditional statistics of the image. In a 3-bit DPCM, the 3-bit error level of each pixel is encoded with a variable-length codeword obtained from a look-up table; which codeword is assigned is determined by the error levels of neighboring, previously transmitted pixels.

At the receiver the error values are decoded by the huffman entropy decoder and the image is decompressed by the 3-bit DPCM decompressor. Pixel interpolation is applied when subsampling was applied to the input image.

2.3 Predictive Vector Quantization (PVQ)

Vector Quantization begins by dividing an image to be transmitted into rectangular blocks of pixels. The transmitter compares each block with a large library of typical blocks, called a "codebook," and selects the library block that best approximates the block to be transmitted. The transmitter then encodes and transmits the index to the selected library block. The receiver, equipped with a copy of the codebook, decodes the index, retrieves the selected library block and inserts it into the output image.

This process is called Vector Quantization because, both theoretically and computationally, each block is treated as a vector. The vector representation of a block can be thought of as laying out all the gray scale values of the block pixels in a single string, that of the upper left pixel first, and of the lower right pixel last. Such a string of numbers comprises

a vector in k-dimensional space, where k is the number of pixels in the block. When the block is treated in this manner, the entire body of mathematical knowledge of vector analysis and multi-dimensional analytical geometry can be brought to bear.

In all the variations of Vector Quantization there is a trade-off between image quality and data compression. In the theoretical limit of zero distortion, the codebook would contain vectors representing all possible blocks. An exact match would always be found. Distortionless transmission would, however, entail an enormous codebook and little data compression, even with optimal coding. At the other extreme, a codebook containing few vectors (representative blocks) would yield large compression ratios, but poor image quality. The objective of any Vector Quantization system design is, therefore, to achieve the best compromise among codebook size, data compression and received image quality.

Predictive vector quantization goes a step further, and consists of a combination of predictive filtering (usually DPCM) and vector quantization[3]. The purpose of the predictive filtering is to remove redundancy before vector quantizing the residue.

2.3.1 Codebook Generation

The codebook generation objective is a codebook that gives low image distortion while minimizing the codebook size. Minimizing the codebook size is important, not only to minimize memory and search time, but also to achieve high compression ratios.

In principle, if one knew the statistics of all images to be transmitted, one could generate a codebook analytically. The most commonly used method consists, however, of using a large number of "training" vectors, where each training vector representing a "typical" image block, to generate the codebook (This study relied upon 12,000 training vectors to generate the codebook). One problem with this approach is that there is some risk that the resulting codebook may be optimal for a few images and far from optimal for others. In effect, the codebook "memorizes" the training images. Thus, the training images must be chosen with care, or other methods must be used to minimize this possibility.

For this report, to control image quality and the degree of compression achievable with PVQ, a Signal to Noise Ratio input was solicitated by the codebook generator. It is expressed

in decibels with reference to the highest gray scale pixel value (255), and the higher the S/N ratio, the better the transmitted image quality (higher number of codebook vectors), but the poorer the compression.

2.3.2 Indices and Codebook Vectors

The transmitted indices consist of codebook vector indices, input vector means, and input vector gains. A vector mean is calculated from an input vector by averaging all the pixel intensities within it, producing a scalar value. A vector gain is a discrete gain level for the input vector obtained from the codebook. Within the codebook, *normalized* codebook vectors are stored by discrete gain levels calculated from the S/N ratio. When the codebook is searched, a high correlation between a *normalized* input vector and codebook vector is determined by taking their dot product (An input vector is normalized by calculating its mean and subtracting the mean from the vector.) and comparing the dot product to the codebook vector's discrete gain level. In general, the lowest gain codebook vector yielding the highest dot product is the codebook vector which most closely matches the input vector.

The receiver uses the vector gain to "roughly" locate, within its copy of the codebook, the codebook vectors which might be used to make the output vector. To find the actual vector, the codebook vector index is used (codebook vector indices are relative to a particular gain). Once the codebook vector is found, it is then multiplied times the gain and summed with the input vector's mean to provide the output vector.

2.3.3 Mean and Gain Processing

When the mean and gain are transmitted, they are sent as separate "images". To gain greater compression these images can be losslessy encoded using BPC in conjunction with arithmetic coding. For this report, the means and gains were sent both with and without additional processing. The codebook indices, in both cases, were left uncoded.

2.4 Transform Coding (Adaptive Discrete Cosine Transform (ADCT))

Transform coding techniques map input imagery into a transform space prior to encoding in the hopes of achieving better compression than might have been achieved if the input imagery itself were compressed. The transformation operation does not provide compression; but, the mapping of the imagery into another domain does permit more effective compression. Better compression can be achieved for two reasons. First, for most applications, not all of the transform domain coefficients need be transmitted to achieve acceptable picture quality. Second, the coefficients that are transmitted can frequently be encoded with reduced precision without seriously affecting image quality.

Generally speaking, transform coding techniques operate as two step processes. In the first step, a linear transformation of the original imagery (separated into sub-blocks of N x N pels) from image space to transform space is performed. In the second step, the transformed image is compressed by encoding each sub-block through quantization and variable length coding. The function of the transformation operation is to make the transformed samples more independent than the original samples, so that the subsequent operation of quantization may be done more efficiently.

Transforms that have proven useful include the Discrete Cosine, Karhunen-Loeve, Walsh-Hadamard, Fourier, Haar, Slant, and Affine transforms. The Karhunen-Loeve transform (KLT) is considered to be an optimum transformation, and for this reason many other transformations have been compared to it in terms of performance.

The Discrete Cosine Transform (DCT) is one of an extensive family of sinusoidal transforms. The DCT has been singled out for special attention by workers in the image processing field, principally because, for conventional imagery having reasonably high inter-element correlation, the DCT's performance is virtually indistinguishable from that of other transforms which are much more complex to implement. In particular, the DCT is one of the best for two important reasons. The first is that it has low susceptibility to the blocking factor.[4] The second is that it comes closest to the KLT[5] in energy compaction[6]; that is, the packing of most of the energy of a block of data into a few uncorrelated coefficients. In addition, the DCT is a fixed transform, known to both transmitter and receiver, and performs almost as well as the KLT. The KLT is a picture-dependent transform which requires intensive computation and the transmission of the transform basis functions for each frame.

The DCT has been chosen by the Joint Photographic Experts Group (JPEG) to be one of the compression algorithms in its "toolkit." JPEG has been working on a still image color compression standard since 1986. This work has its roots in Working Group 8 (WG8) of ISO/IEC JTC1/SC2 (Coded Representation of Picture and Audio Information) which was set up in 1982. In 1985, a CCITT special rapporteur's group was formed to investigate New forms of Image Communication (NIC) under Question 18 in Study Group VIII. The initial work of the NIC group concentrated on a common set of requirements for all the telematic services, including facsimile. JPEG was formed in 1986 by experts from both WG8 and NIC with the expressed goal of selecting a high performance universal compression technique, working under the auspices of WG8. JPEG has recently moved to a new working group - ISO/IEC JTC1/SC2/WG10 (photographic Image Coding). Most of the technical work on JPEG has been completed, and the ISO Committee Draft has been submitted for international balloting.

JPEG's "toolkit" includes algorithms for both progressive and sequential build-up, soft-copy and hard-copy, and a wide range of image compressions. The range includes very-lossy highly-compressed images to lossless (with lower compressions). JPEG bases all of its lossy coding processes on the DCT, and all the lossless processes on a predictive technique. In addition, the DCT as specified by JPEG, can use either Huffman or Arithmetic Coding. This study used the DCT with Arithmetic Coding, and used JPEG's sequential baseline system.

2.4.1 DCT Transformation Process

The DCT and its inverse are formally defined by the pair of equations in Figure 2-4, where f(m,n) (m = row, n = column) are the pixel values in an N by N block, F(u,v) (u, v = horizontal and vertical spatial frequency indices) are the horizontal and vertical spatial frequency components ("coefficients"), and c(u,v) is defined to have the value ½ for u = v = 0, the $\sqrt{½}$ for u = 0 or v = 0, but not both, and 1 for neither u nor v equal to 0.

The forward DCT transforms a square block of image pixel values, typically 8x8, into a similar block of spatial frequency "coefficients." The inverse DCT transforms the coefficients back into the block of image pixels.

Forward DCT

$$F(u,v) = \sum_{i=0}^{7} \sum_{j=0}^{7} f(i,j)\cos(2i+1)u\ pi/16\ \cos(2j+1)v\ pi/16$$

Inverse DCT

$$f(i,j) = \frac{1}{4}\sum_{i=0}^{7} \sum_{j=0}^{7} C(u)C(v)F(u,v)\cos(2i+1)u\ pi/16\ \cos(2j+1)v\ pi/16$$

where

$f(i,j)$: input picture element (from $-2^{(n-1)}$ to $+2^{(n-1)}-1$)

$C(u), C(v)$	$= \sqrt{1/2}$	for $u = 0$ or $v = 0$
	$= 1$	for $u \neq 0$ and $v \neq 0$
	$= 1/2$	for $u = v = 0$

Figure 2-4. Discrete Cosine Transform Equations

2.4.2 Compressing DCT Images

To achieve data compression, one must quantize the coefficients. JPEG recommends that each coefficient be linearly quantized according to a step size assigned to that coefficient, the assigned value being just small enough so that the distortion resulting from quantizing that coefficient is barely noticeable to a human observer. The resulting quantum step numbers are then ranked into an encoding order with the object of placing those quantum numbers most likely to have values of zero last, thus reducing the data to be encoded. JPEG recommends a simple zigzag order that arranges the quantum numbers in order of increasing spatial frequency. This is recommended because most of the energy is contained in the low frequency coefficients with little strength in the high frequency coefficients. To further reduce the data to encode, a threshold is usually established for the quantum numbers and a decision is made to truncate the sequence at the point where the amplitude falls below the chosen threshold. Where the sequence ends is then marked by an end-of-block code word.

The positions and values of the non-zero quantum numbers are then transmitted losslessly by either Huffman coding or Arithmetic coding (This study used Arithmetic coding). The receiver decodes the quantum numbers, multiplies each quantum number by the step size associated with that coefficient to obtain an approximation of the original image. The compression versus distortion trade-off can be controlled by a single quantization scale factor that scales all the step sizes assigned to the coefficients by a single multiplicative constant. The larger this scale factor the greater the compression, but, also, the greater the distortion.

2.4.2.1 Quantization

As mentioned before, after the DCT is applied to the NxN block of pixel values, individual step sizes are used to quantize each spatial frequency coefficient. These step sizes are stored in a *unscaled* quantization matrix. The scale factor is multiplied times this matrix to form a *scaled* quantization matrix. The scaled quantization matrix is then used to quantize the DCT coefficients. In practice, and for computational reasons, the unscaled quantization matrix elements are often multiplied times the scale factor and divided by 50. Any fractional components are discarded. With this in mind, given an unscaled quantization matrix element value of 8 and a scale factor of 55, the resulting scaled quantization matrix element value would be 8 ((8x55)/50=8.8; the fractional .8 is discarded).

Small changes in the scale factor have little or no effect on small valued matrix elements, but do affect larger valued elements. For example, given two unscaled matrix elements with values of 8 and 16, and two scale factors of values 50 and 55, the resulting scaled matrix elements' values would be as follows:

$$(8 \times 50)/50 = 8 \qquad (8 \times 55)/50 = 8$$
$$(16 \times 50)/50 = 16 \qquad (16 \times 55)/50 = 17$$

Thus, small changes in the scaling factor can affect larger matrix element values while leaving smaller matrix element values unchanged.

2.4.2.2 Coefficient Ordering (Ranking)

After quantization, the quantized coefficients are ordered using a simple zigzag sampling pattern which combines them to produce a one-dimensional vector with the coefficients occurring in order of increasing spatial frequency. This principal is illustrated for a 4 by 4 block (See Figure 2-5); actual blocks are 8 by 8 (See Table 2-1). As noted before, the ranking process enhances compression by placing most of the zero-valued quantum numbers last, where they can be ignored, and a end-of-block value often indicates where the last non-zero quantum number is.

Natural Order

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Zigzag Order

1	2	5	9
6	3	4	7
10	13	14	11
8	12	15	16

Figure 2-5. Conversion of Natural Order to Zigzag Order

2.4.2.3 Coding

Starting with the DC coefficient (0), all of the frequency coefficients are coded according to the order of occurrence as shown in the zigzag array. Since many of the high frequency coefficients are zero, a run length coding mechanism and an end-of-block symbol are used to efficiently code runs of zero coefficients.

The quantized frequency coefficients are losslessly encoded using an arithmetic coder and transmitted.

Table 2-1. DCT Coefficient Ordering

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

2.4.2.4 Controlling Image Distortion

Two types of distortion appear in transform coded pictures: truncation error and quantization errors. Quantization errors are noise-like. While truncation errors cause a loss of resolution. In practice, the truncation threshold and

quantization precision must be adjusted experimentally to achieve the maximum compression and acceptable picture quality.

3.0 IMAGE SELECTION

The selection of the images used to compare the four data compression techniques was based on several factors: image quality, image availability, and image content. The images selected were three of the four standard gray scale images previously developed by DIS for the NCS in prior studies. These images exhibit characteristics which thoroughly test gray scale data compression techniques. In addition, to compensate for any techniques which favor a particular image resolution, higher resolutions of two of the three images were also used. Thus, in total, five images, three low resolution and two high resolution, were used to compare the four data compression techniques.

The inherent characteristics of these images, in addition to thoroughly testing the gray scale data compression techniques, also aid in the subjective evaluation of the resulting output images. For example, the IEEE image is representative of an identification card, combining both photographic and textual information, and includes a high contrast wedge that aids in the evaluation of an algorithm's effect on resolution. The house and sky image contains large areas of gradually changing gray scale, several areas of varying texture, and various horizontal, vertical, and diagonal lines. The house with trees image is similar, but also contains regions of high detail. Of these three, the two chosen for high resolution comparisons were the IEEE image and the house with trees image.

4.0 SUBJECTIVE MEASUREMENT OF CODER PERFORMANCE

To subjectively measure the quality of the images provided by the four selected data compression techniques, their output images should be the decompressed result of compressing an original image to the same compression level for all four techniques. Unfortunately, this is impractical. For example, DPCM and Bit Plane Coding provide only discrete and *unequal* levels of compression. In some cases, these levels of compression are close in value, but since they are unequal, and since it is unknown as to what difference in compression between the four techniques is perceptible to the human eye, they can not be considered equal for the sake of visual image comparisons. Therefore, to elicit which compression technique(s), if any, provides equal or better image quality with greater compression than the other techniques, their compression inequalities must be considered.

The compression inequalities can be used to advantage. Typically, with lossy compression techniques (which all four are), the higher the compression, the poorer the image quality. Thus, by comparing image quality versus compression, one can determine which compression technique provides better image quality for a given compression level. For example, given two output images compressed by two different compression techniques with different but nearly equal levels of compression, if the image with the higher level of compression exhibits equal or better image quality than the image with the lower level of compression, then one can conclude that the compression technique which produced the image with the higher level of compression provides the better image quality. The reverse, however, is not true. Higher quality images coupled with lower compressions as compared to lower quality images with higher compressions merely reinforces the original premise: image quality degrades with higher compressions for lossy compression techniques. The purpose of this study, given the above, was to determine if any of the selected data compression techniques provides superior image quality, or if image quality degrades more slowly for one than for the others as compression is increased.

4.1 Subjective Evaluation Environment

To rank the four selected data compression techniques, an image viewing and comparison room was established to compare selected output images generated by the four data compression techniques. In the viewing room, up to seven subjects could view and evaluate the data compression techniques' output images on a 512 pixel by 512 pixel video monitor. Although more subjects could have viewed the monitor simultaneously, a smaller number ensured that all subjects had a clear and unobstructed view for making fair comparisons. Altogether, by convening three separate panels of subjects, a total of nineteen subjects evaluated and compared the output images.

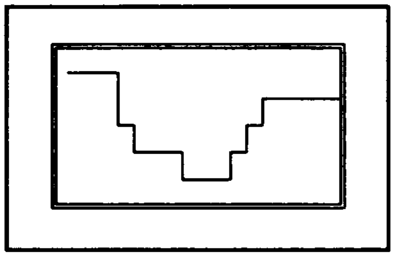

Figure 4-1. Whole Scene

To simplify the subjects' task of comparing the images, two images were shown side by side on the monitor, and the subjects were asked to select which image exhibited better image quality. They were asked to do this even if the images appeared equal in quality. Since the resolving capability and viewing size of the monitor did not permit showing two whole images side by side, the images were halved, and the subjects were asked to compare either the left versus the left halves, the right versus the right halves, or the left versus the right halves (See Figure 4-1 through Figure 4-3). Which side a data compression technique's output image was placed on (left or right) was done by random selection by a computer.

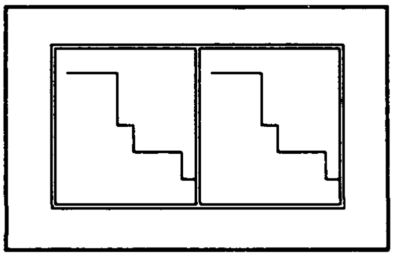

Figure 4-2. Left Versus Left

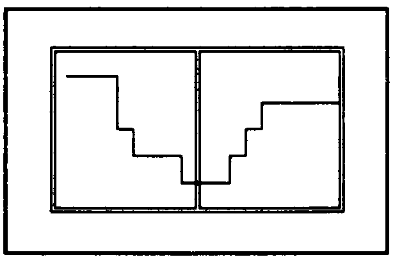

Figure 4-3. Left Versus Right

For the high resolution images, this method of image comparison, half on the left and half on the right, was not directly applicable. When displayed on the monitor, the high resolution images are magnified 2:1. So, smaller portions of the images were compared.

In addition, as a control, data compression techniques' output images were compared against themselves, the original images were compared against themselves, and the original images were compared against data compression techniques' output images. In these cases, the

outcomes were expected to be equal (or nearly so), or, in the case of an original image versus a data compression technique's output image, the original image was expected to be preferred over the data compression technique's output image (unless the output image quality approached the original image, then they could have been chosen equally).

To ensure that the comparisons made by the subjects were fair, only subjects who were "unexperienced" viewers were chosen. To qualify as an unexperienced viewer, a subject had to lack experience and knowledge of the visual effect(s) of any of the data compression techniques on image quality.

Each subject on a panel was provided with evaluation materials (See Figure 4-4 and Figure 4-5) consisting of a brief statement which described what was expected, and an image comparison score card. In addition, the subjects were not allowed to discuss the images, nor were they allowed to compare their selections.

IMAGE COMPARISON

Participant's Role

As a participant, you will be shown 70 split-screen images on a video monitor. The images consist of a left and right portion. You must decide which portion, left or right, exhibits the better image quality. A choice must be made regardless of how close the image quality of the two portions may be. Mark your choice on the appropriate spot on the comparison sheet. Please note that the left and right portions may show the same scene (See Figure 2), or may show the left and right portions of a particular scene (See Figure 3).

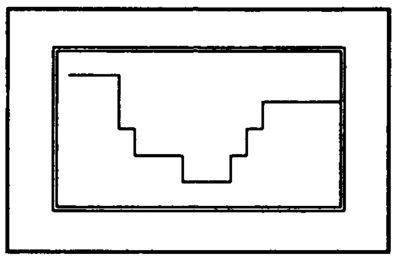

Figure 1. Scene Example

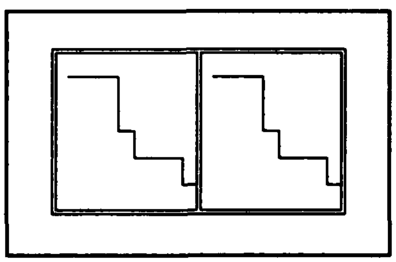

Figure 2. Same Scene on Left and Right Portions

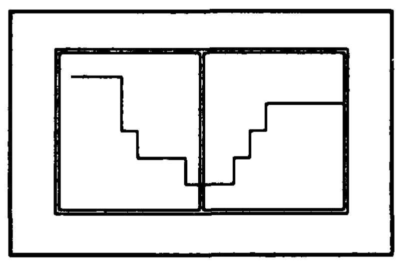

Figure 3. Left and Right Portions Showing Whole Scene

Figure 4-4. Subject Viewing Instructions

Date ________ *Time* ________ *Name* ______________

IMAGE COMPARISON

(Session 1)

Number	Left	Right
01	___	___
02	___	___
03	___	___
04	___	___
05	___	___
06	___	___
07	___	___
08	___	___
09	___	___
10	___	___
11	___	___
12	___	___
13	___	___
14	___	___
15	___	___
16	___	___
17	___	___
18	___	___
19	___	___
20	___	___
21	___	___
22	___	___
23	___	___
24	___	___
25	___	___
26	___	___
27	___	___
28	___	___
29	___	___
30	___	___
31	___	___
32	___	___
33	___	___
34	___	___
35	___	___

Figure 4-5a. Image Comparison Score Card

Date ________ *Time* _______ *Name* ____________

IMAGE COMPARISON

(Session 2)

Number	Left	Right
36	____	____
37	____	____
38	____	____
39	____	____
40	____	____
41	____	____
42	____	____
43	____	____
44	____	____
45	____	____
46	____	____
47	____	____
48	____	____
49	____	____
50	____	____
51	____	____
52	____	____
53	____	____
54	____	____
55	____	____
56	____	____
57	____	____
58	____	____
59	____	____
60	____	____
61	____	____
62	____	____
63	____	____
64	____	____
65	____	____
66	____	____
67	____	____
68	____	____
69	____	____
70	____	____

Figure 4-5b. Image Comparison Score Card (Continued)

4.2 Evaluation Combinations

The number of output image combinations viewed by the three panels were limited to 70 combinations, and, as noted before, contained comparisons between output images for different compression techniques, between output images for a compression technique and the original image, between the same output image for a compression technique, and between the same original image. The last three comparison types were used as a control to measure noise or variability in the panels' decisions.

All information needed to compare the four data compression techniques came from the first comparison type: comparisons between output images from different compression techniques. In theory, these comparisons were to be done for two different resolutions, for three images at a low resolution and two images at a high resolution, and for three different compression levels (low, medium, high). In addition, the comparisons must determine if an output image from a compression technique using a higher compression than another technique also provides an equal or better quality image. (Please note that only if the higher compression algorithm is preferred is the result significant; selecting the lower compression algorithm is not.) Altogether, the number of possible combinations coupled with the control measures (301 combinations) was too high for efficient viewing by the evaluation panels.

The number of combinations was calculated as follows. Given the four data compression techniques where one is compared against the other three, one at a time, there are 12 possible combinations[7]:

$$P(n,r) = n!/(n-r)!$$

where:

$P(n,r)$	is the number of possible combinations
n	is the number of data compression techniques
r	is the number of output images viewed at one time

given:

$$n = 4$$
$$r = 2$$

then:

$$P(n,r) = 4!/(4-2)! = 4!/2! = 24/2 = 12$$

Taken in conjunction with three images at one resolution and three compression levels, and two images at a higher resolution and three compression levels, there are a total of 180 possible combinations without including control measures:

$$P = \text{algorithm combinations} \times (\#\ \text{compression levels} \times (\text{number images}))$$

or

$$P = 12 \times (3 \times (3 + 2)) = 12 \times (3 \times 5) = 12 \times 15 = 180$$

Control measures increase the number of possible combinations by 121 for a total of 301 combinations:

$$P_c = (\text{Number of algorithms} \times (\#\ \text{compression levels} \times (\text{number images} \times (\text{original} + \text{self})))) + \text{original versus original}$$

or

$$P_c = (4 \times (3 \times (5 \times (1 + 1)))) + 1 = (4 \times (3 \times (10))) + 1 = (4 \times 30) + 1 = 121$$

thus

$$P_t = P + P_c = 180 + 121 = 301$$

To reduce the number of combinations to be viewed by the evaluation panels, 70 image comparisons were selected at random from these 301 combinations using a computerized random

selection. Although the 70 combinations selected are a subset of all combinations, it is a large enough set to provide sufficient information to compare the four data compression techniques with regards to image quality at a particular compression level.

The images selected for comparison were picked from a suite of images which were the result of coding the five originals to different compression levels using the four compression techniques. Table 4-2 shows the suite of images, and Table 4-1 shows the low, medium, and high compression level ranges chosen for them.

Table 4-1. Compression Level Assignments

		Compression Technique			
		DPCM (Filter)	BPC (# Bit Planes)	PVQ (S/N Ratio)	ADCT (Scale Factor)
Compression Level	Low	None	6-bit	50	7-25
	Medium	1-Dimensional	3,4-bit	30-40	32-59
	High	2-Dimensional	2-bit	20-25	100-200

Since the images are being shown side by side on a monitor, and the monitor could bias the opinions of the evaluation panels if one side of the monitor's screen displays images differently as compared to the other side, a few of the comparisons were repeated with the images swapped. For example, the image on the left side was moved to the right side, and the image on the right side was moved to the left side. The combinations chosen for evaluation are shown in Table 4-3. Please note that the entries marked with a "†" are the control combinations. In addition, the (L,L), (L,R), (R,L), and (R,R) following the compression techniques' names indicates the side of the monitor screen (left or right) a technique's output image was displayed on (the first L or R) and what portion of the output image (left or right) was displayed (the second L or R).

Table 4-2. Compressed Image Suite

Compression Technique			200 pixel per inch images: House with sky	200 pixel per inch images: House with trees	200 pixel per inch images: Face	400 pixel per inch images: House with trees	400 pixel per inch images: Face
DPCM	No Filtering		1.32	1.93	1.26	**1.61**	1.18
DPCM	1-Dimensional Filtering		**0.70**	**1.07**	**0.67**	0.93	0.61
DPCM	2-Dimensional Filtering		**0.39**	**0.56**	**0.39**	-	-
BPC	6-bit		**1.96**	-	2.58	-	1.05
BPC	4-bit		**0.78**	2.33	**1.11**	**1.85**	-
BPC	3-bit		0.56	**1.74**	**0.85**	**1.31**	**0.52**
BPC	2-bit		**0.32**	**1.10**	**0.53**	**0.81**	**0.30**
PVQ	Coding	S/N Ratio					
PVQ	No coding of Bit Planes	50	**1.23**	**1.56**	**1.29**	1.60	1.29
PVQ	No coding of Bit Planes	40	1.10	**1.34**	1.11	1.30	**1.11**
PVQ	No coding of Bit Planes	30	1.02	**1.09**	1.02	1.09	1.02
PVQ	No coding of Bit Planes	25	-	-	-	1.06	1.01
PVQ	No coding of Bit Planes	20	-	-	-	1.06	1.00
PVQ	Arithmetic Coding of Bit Planes	50	0.74	1.28	0.87	1.06	0.78
PVQ	Arithmetic Coding of Bit Planes	40	0.60	1.12	**0.67**	**1.02**	0.52
PVQ	Arithmetic Coding of Bit Planes	30	0.45	0.78	0.50	0.65	0.39
PVQ	Arithmetic Coding of Bit Planes	25	0.41	**0.60**	0.45	0.56	-
PVQ	Arithmetic Coding of Bit Planes	20	0.40	**0.58**	**0.43**	0.52	0.38
ADCT	Scale Factor						
ADCT	7		-	-	-	-	**1.13**
ADCT	9		**1.29**	-	-	-	-
ADCT	10		-	-	**1.23**	-	-
ADCT	11		-	-	-	1.78	-
ADCT	15		-	**1.92**	-	-	-
ADCT	16		-	-	-	-	0.68
ADCT	25		**0.68**	1.46	**0.72**	-	-
ADCT	32		-	-	-	**1.08**	-
ADCT	33		-	-	-	-	**0.44**
ADCT	50		0.44	**1.01**	0.47	**0.84**	0.33
ADCT	58		-	-	0.44	-	-
ADCT	59		**0.40**	-	-	-	-
ADCT	100		0.27	0.67	0.31	0.56	0.21
ADCT	131		-	**0.57**	-	-	-
ADCT	200		0.17	**0.41**	0.20	0.35	0.13

Table 4-3. Combination of Algorithms Selected for Comparison

Number	Image	Resolution (pels/inch)	Compression Level	Compression Technique with Higher Compression: Compression Technique	bits/ pixel	Compression Technique with Lower Compression: Compression Technique	bits/ pixel
01	House with trees	200	medium	DPCM (L,L)	1.07	ADCT (R,L)	1.92
02	Face	200	high	PVQ (R,L)	.43	ADCT (L,L)	.47
03	House with trees	400	high	BPC (R,R)	.81	DPCM (L,R)	.93
04	House with trees	400	medium	PVQ (L,L)	1.02	DPCM (R,L)	1.61
05	House with sky	200	medium	DPCM (L,L)	.70	PVQ (R,L)	1.23
06	House with trees	400	medium	PVQ (R,R)	1.02	ADCT (L,R)	1.08
07	House with sky	200	medium	ADCT (L,R)	.68	PVQ (R,R)	1.23
08	House with trees	400	medium	ADCT (L,R)	.84	BPC (R,R)	1.31
09	Face	200	low	ADCT (L,R)	1.23	DPCM (R,R)	1.26
10	House with trees	400	medium	ADCT (R,L)	.84	PVQ (L,L)	1.02
11	Face	200	low	ADCT (L,L)	1.23	BPC (R,L)	2.58
12	House with trees	400	medium	BPC (R,R)	1.31	PVQ (L,R)	1.60
13	House with sky	200	medium	DPCM (L,L)	.70	BPC (R,L)	.78
14	House with trees	200	low	PVQ (R,R)	1.56	DPCM (L,R)	1.93
15	House with sky	200	low	ADCT (L,L)	1.29	DPCM (R,L)	1.32
16	House with trees	400	low	PVQ (R,L)	1.60	DPCM (L,L)	1.61
17	House with sky	200	high	DPCM (R,R)	.39	ADCT (L,R)	.44
18	Face	200	high	BPC (L,L)	.53	DPCM (R,L)	.67
19	Face	200	high	PVQ (L,R)	.43	DPCM (R,R)	.67
20	House with sky	200	medium	BPC (L,R)	.78	ADCT (R,R)	1.29
21	Face	400	low	ADCT (L,L)	1.13	DPCM (R,L)	1.18
22	Face	400	medium	ADCT (L,L)	.44	BPC (R,L)	.52
23	Face	400	medium	BPC (L,R)	.52	ADCT (R,R)	.68
24	House with trees	200	high	ADCT (L,R)	.41	BPC (R,R)	1.10
25	House with trees	200	low	ADCT (R,R)	1.92	BPC (L,R)	2.33
26	Face	400	medium	ADCT (R,L)	.44	PVQ (L,L)	.52
27	House with trees	200	medium	ADCT (R,L)	1.01	PVQ (L,L)	1.12
28	Face	400	high	BPC (L,R)	.30	ADCT (R,R)	.33
29	Face	400	medium	PVQ (L,L)	1.11	DPCM (R,L)	1.18
30	House with trees	200	high	DPCM (L,R)	.56	ADCT (R,R)	.67
31†	Face	200	low	PVQ (L,L)	1.29	Original (R,R)	-
32	Face	200	high	BPC (R,R)	.53	DPCM (L,L)	.67
33†	House with trees	200	medium	ADCT (L,L)	1.01	Original (R,R)	-
34	Face	200	low	ADCT (L,L)	1.23	PVQ (R,R)	1.28
35	Face	200	high	DPCM (R,R)	.39	PVQ (L,L)	.43
36	Face	200	high	DPCM (L,L)	.39	BPC (R,R)	.53
37†	House with trees	200	high	PVQ (R,R)	.58	Original (L,L)	-
38	Face	200	medium	PVQ (R,R)	.67	ADCT (L,L)	.72
39†	House with trees	200	medium	BPC (L,L)	1.74	BPC (R,R)	1.74

Number	Image	Resolution (pels/inch)	Compression Level	Compression Technique with Higher Compression		Compression Technique with Lower Compression	
				Compression Technique	bits/ pixel	Compression Technique	bits/ pixel
40†	Face	200	medium	ADCT (R,R)	.72	Original (L,L)	-
41	House with sky	200	medium	ADCT (L,L)	.68	DPCM (R,R)	.70
42†	Face	200	high	DPCM (L,L)	.39	Original (R,R)	-
43†	House with trees	200	high	PVQ (L,L)	.58	Original (R,R)	-
44†	Face	400	low	ADCT (L,L)	1.13	ADCT (R,R)	1.13
45†	House with sky	200	medium	ADCT (L,L)	.40	ADCT (R,R)	.40
46†	House with sky	200	low	BPC (R,R)	1.96	Original (L,L)	-
47	House with trees	200	high	BPC (R,R)	1.10	PVQ (L,L)	1.12
48†	House with trees	400	medium	BPC (L,L)	1.85	Original (R,R)	-
49	Face	200	medium	BPC (L,L)	1.11	ADCT (R,R)	1.28
50	Face	200	medium	BPC (L,L)	1.11	DPCM (R,R)	1.26
51	Face	200	medium	ADCT (L,L)	.72	BPC (R,R)	1.19
52	House with sky	200	low	PVQ (L,L)	1.23	DPCM (R,R)	1.32
53	House with trees	200	medium	DPCM (R,R)	1.07	BPC (L,L)	1.10
54	House with trees	200	high	ADCT (R,R)	.57	BPC (L,L)	1.10
55	House with sky	200	low	PVQ (L,L)	1.23	BPC (R,R)	1.96
56	Face	200	medium	BPC (R,R)	1.11	ADCT (L,L)	1.23
57	Face	200	high	DPCM (R,R)	.39	ADCT (L,L)	.44
58	House with trees	200	high	PVQ (R,R)	.60	BPC (L,L)	1.10
59	House with trees	200	high	PVQ (L,L)	.58	BPC (R,R)	1.10
60	House with sky	200	high	BPC (R,R)	.32	DPCM (L,L)	.39
61	House with trees	200	high	BPC (L,L)	1.10	PVQ (R,R)	1.12
62†	House with trees	200	high	BPC (L,L)	1.10	Original (R,R)	-
63†	Face	200	medium	DPCM (R,R)	.67	Original (L,L)	-
64†	Face	200	medium	BPC (L,L)	.85	Original (R,R)	-
65	Face	200	high	DPCM (L,L)	.39	PVQ (R,R)	.43
66†	House with trees	200	medium	ADCT (L,L)	.57	ADCT (R,R)	.57
67	House with trees	200	high	DPCM (R,R)	.56	PVQ (L,L)	.58
68†	House with trees	200	medium	ADCT (R,R)	.57	Original (L,L)	-
69†	Face	200	high	DPCM (R,R)	.39	Original (L,L)	-
70†	Face	200	n/a	Original (L,L)		Original (R,R)	-

5.0 RESULTS AND ANALYSIS OF SUBJECTIVE COMPARISON

As noted before, to determine which data compression technique(s) exhibit the best image quality retention as compression increases, if an output image from one technique exhibited equal or better image quality at a higher compression level than another technique, then the former does exhibit better image quality retention than the latter. The results from the subjective comparison of the output images of the four data compression techniques indicate that some of the techniques do exhibit better image quality retention than other techniques. The raw results of the comparison are shown in Table 5-[illegible]se note that the gray-shaded combinations are the comparisons which indicate [illegible]ge quality retention for a technique with a higher compression value.

Table 5-1. Image Comparison Scoring

Number	Image	Resolution (pels/inch)	Compression Level	Compression Technique with Higher Compression Value: Algorithm	Score	Compression Technique with Lower Compression Value: Algorithm	Score
01	House with trees	200	medium	DPCM (L,L)	00	ADCT (R,L)	19
02	Face	200	high	PVQ (R,L)	00	ADCT (L,L)	19
03	House with trees	400	high	BPC (R,R)	00	DPCM (L,R)	19
04	House with trees	400	medium	PVQ (L,L)	09	DPCM (R,L)	10
05	House with sky	200	medium	DPCM (L,L)	03	PVQ (R,L)	16
06	House with trees	400	medium	PVQ (R,R)	02	ADCT (L,R)	17
07	House with sky	200	medium	ADCT (L,R)	17	PVQ (R,R)	02
08	House with trees	400	medium	ADCT (L,R)	18	BPC (R,R)	01
09	Face	200	low	ADCT (L,R)	13	DPCM (R,R)	06
10	House with trees	400	medium	ADCT (R,L)	10	PVQ (L,L)	09
11	Face	200	low	ADCT (L,L)	11	BPC (R,L)	08
12	House with trees	400	medium	BPC (R,R)	01	PVQ (L,R)	18
13	House with sky	200	medium	DPCM (L,L)	16	BPC (R,L)	03
14	House with trees	200	low	PVQ (R,R)	01	DPCM (L,R)	18
15	House with sky	200	low	ADCT (L,L)	05	DPCM (R,L)	14

Number	Image	Resolution (pels/inch)	Compression Level	Compression Technique with Higher Compression Value: Algorithm	Score	Compression Technique with Lower Compression Value: Algorithm	Score
16	House with trees	400	low	PVQ (R,L)	03	DPCM (L,L)	16
17	House with sky	200	high	DPCM (R,R)	00	ADCT (L,R)	19
18	Face	200	high	BPC (L,L)	00	DPCM (R,L)	19
19	Face	200	high	PVQ (L,R)	00	DPCM (R,R)	19
20	House with sky	200	medium	BPC (L,R)	04	ADCT (R,R)	15
21	Face	400	low	ADCT (L,L)	12	DPCM (R,L)	07
22	Face	400	medium	ADCT (L,L)	19	BPC (R,L)	00
23	Face	400	medium	BPC (L,R)	01	ADCT (R,R)	18
24	House with trees	200	high	ADCT (L,R)	18	BPC (R,R)	01
25	House with trees	200	low	ADCT (R,R)	11	BPC (L,R)	08
26	Face	400	medium	ADCT (R,L)	16	PVQ (L,L)	03
27	House with trees	200	medium	ADCT (R,L)	17	PVQ (L,L)	02
28	Face	400	high	BPC (L,R)	00	ADCT (R,R)	19
29	Face	400	medium	PVQ (L,L)	01	DPCM (R,L)	18
30	House with trees	200	high	DPCM (L,R)	00	ADCT (R,R)	19
31†	Face	200	low	PVQ (L,L)	07	Original (R,R)	12
32	Face	200	high	BPC (R,R)	00	DPCM (L,L)	19
33†	House with trees	200	medium	ADCT (L,L)	10	Original (R,R)	09
34	Face	200	low	ADCT (L,L)	10	PVQ (R,R)	09
35	Face	200	high	DPCM (R,R)	18	PVQ (L,L)	01
36	Face	200	high	DPCM (L,L)	19	BPC (R,R)	00
37†	House with trees	200	high	PVQ (R,R)	00	Original (L,L)	19
38	Face	200	medium	PVQ (R,R)	03	ADCT (L,L)	16
39†	House with trees	200	medium	BPC (L,L)	13	BPC (R,R)	06
40†	Face	200	medium	ADCT (R,R)	11	Original (L,L)	08
41	House with sky	200	medium	ADCT (L,L)	18	DPCM (R,R)	01
42†	Face	200	high	DPCM (L,L)	02	Original (R,R)	17
43†	House with trees	200	high	PVQ (L,L)	00	Original (R,R)	19
44†	Face	400	low	ADCT (L,L)	10	ADCT (R,R)	09

Number	Image	Resolution (pels/inch)	Compression Level	Compression Technique with Higher Compression Value		Compression Technique with Lower Compression Value	
				Algorithm	Score	Algorithm	Score
45†	House with sky	200	medium	ADCT (L,L)	09	ADCT (R,R)	10
46†	House with sky	200	low	BPC (R,R)	10	Original (L,L)	09
47	House with trees	200	high	BPC (R,R)	00	PVQ (L,L)	19
48†	House with trees	400	medium	BPC (L,L)	04	Original (R,R)	15
49	Face	200	medium	BPC (L,L)	00	ADCT (R,R)	19
50	Face	200	medium	BPC (L,L)	00	DPCM (R,R)	19
51	Face	200	medium	ADCT (L,L)	19	BPC (R,R)	00
52	House with sky	200	low	PVQ (L,L)	04	DPCM (R,R)	15
53	House with trees	200	medium	DPCM (R,R)	18	BPC (L,L)	01
54	House with trees	200	high	ADCT (R,R)	18	BPC (L,L)	01
55	House with sky	200	low	PVQ (L,L)	04	BPC (R,R)	15
56	Face	200	medium	BPC (R,R)	01	ADCT (L,L)	18
57	Face	200	high	DPCM (R,R)	04	ADCT (L,L)	15
58	House with trees	200	high	PVQ (R,R)	13	BPC (L,L)	06
59	House with trees	200	high	PVQ (L,L)	15	BPC (R,R)	04
60	House with sky	200	high	BPC (R,R)	02	DPCM (L,L)	17
61	House with trees	200	high	BPC (L,L)	01	PVQ (R,R)	18
62†	House with trees	200	high	BPC (L,L)	01	Original (R,R)	18
63†	Face	200	medium	DPCM (R,R)	07	Original (L,L)	12
64†	Face	200	medium	BPC (L,L)	00	Original (R,R)	19
65	Face	200	high	DPCM (L,L)	19	PVQ (R,R)	00
66†	House with trees	200	medium	ADCT (L,L)	09	ADCT (R,R)	10
67	House with trees	200	high	DPCM (R,R)	18	PVQ (L,L)	01
68†	House with trees	200	medium	ADCT (R,R)	04	Original (L,L)	15
69†	Face	200	high	DPCM (R,R)	03	Original (L,L)	16
70†	Face	200	n/a	Original (L,L)	12	Original (R,R)	07

5.1 Bias Detection

To detect bias in the comparisons, like defects in the monitor screen (left versus right), or preferences for either the left side or right side of the screen by the subjects, some images were compared to themselves. In total, four 200 pel/inch and one 400 pel/inch comparisons were made (See Table 5-2).

Table 5-2. Images Compared Against Themselves

Number	Image	Resolution (pels/inch)	Compression Level	Compression Technique with Higher Compression Value: Algorithm	Score	Compression Technique with Lower Compression Value: Algorithm	Score
39	House with trees	200	medium	BPC (L,L)	13	BPC (R,R)	06
44	Face	400	low	ADCT (L,L)	10	ADCT (R,R)	09
45	House with sky	200	medium	ADCT (L,L)	09	ADCT (R,R)	10
66	House with trees	200	medium	ADCT (L,L)	09	ADCT (R,R)	10
70	Face	200	n/a	Original (L,L)	12	Original (R,R)	07
Total Score					53		42

Without bias there is an equal probability of either side being selected. Table 5-3 shows for nineteen trials the probabilities of all possible outcomes for two equally likely events. The values in the table were calculated for two mutually exclusive and equally likely events using a binomial distribution.[8] From it, if a 95 percent confidence level is desired for the outcome distribution of 19 trials for two equally likely events, then for all trials an event should have 6 to 13 successes (Gray-shaded areas). All five of the same image comparisons fall in this range. Nevertheless, for all five comparisons, the probability of either side being chosen should have been closer to 0.5. Using a binomial distribution, the probability of obtaining a score of 42 or less for the right side of an image on the

Table 5-3. Binomial Distribution for 19 Trials for Two Events of Equal Probability

19 Trials with 0.5 Probability of Success			
Probability of n or less successes		Probability of n or more successes	
n	P	n	P
0	.0000019	10	.500
1	.000038	11	.324
2	.00036	12	.180
3	.00221	13	.0835
4	.00961	14	.0318
5	.0318	15	.00961
6	.0835	16	.00221
7	.180	17	.00036
8	.324	18	.000038
9	.500	19	.0000019

right side of the screen is 0.15. Thus, we can conclude that there may be a bias favoring either the left side of the screen, or the left side of the image. (for all five comparisons, the left side of the image was on the left side of the screen, and the right side of the image was on the right side of the screen). Whether this bias significantly affected the results will be discussed later.

5.2 Comparison Analysis

Again using Table 5-3 with a 95 percent confidence level, for the comparisons where output images from different compression techniques were compared, if the output image for the compression technique achieving a higher compression value was give an objective score in the range of 6 to 13, then its image can be considered equal in quality to the output image of the compression technique which achieved a lower compression value. Secondly, if the former achieved a score greater than 13, then its output image can be considered to be of greater quality than the latter's. For scores of 5 or less, however, no conclusions can be drawn; they merely state that the quality of the output image for the technique with the higher compression was not equal or better than the output image for the technique with the lower compression. The scores for the comparisons between the four compression techniques are shown in Table 5-4.

In general, Table 5-4 shows the four compression techniques have the following ranking:

1. ADCT (best)
2. DPCM
3. PVQ
4. BPC (worst)

Please note that in the table, for some combinations there may be more than one entry. For these, the entries appearing above the line are scores obtained for an image comparison relevant to that combination. The entry below the line is the mean of all entries (or scores) appearing above the line for that combination.

Of the four compression techniques, ADCT was, by far, considered better at providing better or equal quality images at higher compressions by the three subject panels. Indeed, the higher the compression the more ADCT's output images were preferred over the other three. Also, there is no conclusive evidence contrary to the selected ranking. For example, where

Table 5-4. Tabulated Results of Subjective Comparison

Compression technique with higher compression value	Compression technique with lower compression value	Subjective Mean Opinion Score in Favor of Compression Technique with Higher Compression Value					
		200 pels/inch Compression Levels			400 pels/inch Compression Levels		
		Low	Medium	High	Low	Medium	High
DPCM	BPC	-	16.0 18.0 17.0	19.0	-	-	-
	PVQ	-	3.0	18.0 19.0 18.0 18.3	-	-	-
	ADCT	-	0.0	0.0 0.0 4.0 1.3	-	-	-
BPC	DPCM	-	0.0	0.0 0.0 2.0 1.3	-	-	0.0
	PVQ	-	-	0.0 1.0 0.5	-	1.0	-
	ADCT	-	4.0 0.0 1.0 1.7	-	-	1.0	0.0
PVQ	DPCM	1.0 4.0 2.5	-	0.0	3.0	9.0 1.0 5.0	-
	BPC	4.0	-	13.0 15.0 14.0	-	-	-
	ADCT	-	3.0	0.0	-	2.0	-
ADCT	DPCM	13.0 5.0 9.0	18.0	-	12.0	-	-
	BPC	11.0 11.0 11.0	19.0	18.0 18.0 18.3	-	18.0 19.0 18.5	-
	PVQ	10.0	17.0 17.0 17.0	-	-	10.0 16.0 13.0	-

output images from ADCT were judged equal or better than output images from VQ, output images from VQ were not judged equal or better than output images from ADCT.

5.3 Effect of Bias

Previously, it was determined that bias in favor of the left side of the monitor screen or the left side of an image may have been present in the subjective comparison. Nevertheless, additional analysis shows that the potential bias failed to significantly affect the results, or the subsequent algorithm rankings.

To ascertain that the potential bias failed to significantly affect the results and rankings, if one or more of the comparison results for an algorithm ranking, as compared to the next lower ranked algorithm, was significant (score of 14 or higher), and was for the *right* side of an image appearing on the *right* side of the monitor, then the ranking of those two algorithms is correct. For DPCM (higher ranking) versus PVQ (lower ranking), this is the case. For two out of three comparisons, DPCM was on the right side of the screen and was used on the right side of the image, and received a score of 18 for both. Thus, the ranking of DPCM higher than PVQ is correct.

For PVQ versus BPC, it is not quite so clear cut. For them, for one out of two comparisons, PVQ received a score of 13 for the right side of an image on the right side of the screen. This score falls just within the 95 percent range where images are considered equal, and therefore can not be considered significant. Nevertheless, by using the aggregate of the two comparisons, the bias can be ignored, and if the aggregate score in favor of PVQ is significant, then PVQ can be ranked higher than BPC. The bias can be ignored because PVQ was used on both sides of the screen and on both sides of the image (left side of image on left side of screen, and right side of image on right side of screen). Subsequently, when the scores for PVQ are summed, the bias negates itself. For the aggregate score of both comparisons, PVQ received a total score of 28 out of 38 with a probability of 0.00255, a significant result (as judged by the less than 5% range). Thus, PVQ can be ranked higher than BPC.

For ADCT versus DPCM, there is only one significant comparison result to examine, and, unfortunately, it used DPCM on the favorably biased side of the screen. To determine if this one result is truely significant, an estimate of the potential bias is needed. The worst case for the bias can be estimated by ascertaining what the magnitude of the bias should be to make a score of 18, in favor of the left side of the screen, non-significant. Theoretically, if the image on both sides of the screen are picked equally, then the probability of either side being selected

should be 0.5. With a bias in favor of the left side of the screen these probabilities would have to shift from 0.5 for both sides to 0.77 and 0.23 for the left and right sides of the screen, respectively, to cause a score of 18 for the left side of the screen to become non-significant. If the potential bias is less than the worst case bias, then the score of 18 that ADCT received is significant, and ADCT can be ranked higher than DPCM. To determine if the potential bias is less, the scoring results for the five identical image comparisons can be used. Given the worst case bias, if the probability of getting a score of 42 out of 95 for the right side is very low, then the potential bias is less than the worst case bias, and ADCT can be ranked higher than DPCM. This is the case. The probability of getting a score of 42 out of 95 for the right side of the screen, given the worst case bias, is 0.000004. Thus, ADCT's score of 18 versus 1 for DPCM is significant, and ADCT can be ranked higher than DPCM.

5.4 Comparison Between Objective and Subjective Measurements

Typically, to objectively measure the difference in quality between an algorithm's output images and the original image, the Root Mean Square (RMS) of the number of differences between an output image and the original is used[9],[10],[11]. Nevertheless, the RMS error is not necessarily an accurate yardstick. For example, it is possible for an output image from one compression technique to have a higher RMS error value than an output image from another compression technique, but yet be more pleasing to the eye, and thus be considered to be of better quality.

A comparison of the objective and subjective measurements indicates that this is the case. For some output image comparisons between two different compression techniques, the image with the higher RMS error value was found by the three panels to be of equal or higher quality than the image with the lower RMS value (Gray-shaded rows in Table 5-5; asterisks next to the number indicate those comparisons where images with higher compression were judged equal or better in quality than those with lower compression). Of the output images compared, there were 14 comparisons where the RMS error values were available. Of these fourteen, eight subjective image comparisons (gray shaded rows), or fifty-seven percent of the comparisons, failed to agree with the objective RMS error values.

For these comparisons, please note that the output images with the higher RMS coupled with the higher subjective score (the eight comparisons which disagree on a subjective and

objective basis) were processed by the two highest ranked compression algorithms. In fact, of the eight comparisons, ADCT accounted for four, and DPCM accounted for four, an even split. One conclusion which might be made from these results is that the ADCT and DPCM algorithms were better able to disguise errors in their output images than the other two compression techniques, thus yielding, at least to the human eye, better image quality at a given compression level.

Table 5-5. Comparison of Objective and Subjective Measurements

Number	Image	Resolution (pels/inch)	Compression Technique with Higher Compression Value: Algorithm	Score	RMS Error	Compression Technique with Lower Compression Value: Algorithm	Score	RMS Error
02	Face	200	PVQ (R,L)	00	2.16	ADCT (L,L)	19	6.8
05	House with sky	200	DPCM (L,L)	03	3.88	PVQ (R,L)	16	2.29
07*	House with sky	200	ADCT (L,R)	17	1.85	PVQ (R,R)	02	2.29
14	House with trees	200	PVQ (R,R)	01	7.46	DPCM (L,R)	18	6.7
17	House with sky	200	DPCM (R,R)	00	2.5	ADCT (L,R)	19	5.53
19	Face	200	PVQ (L,R)	00	6.80	DPCM (R,R)	19	9.74
27*	House with trees	200	ADCT (R,L)	17	6.93	PVQ (L,L)	02	5.41
30	House with trees	200	DPCM (L,R)	00	13.5	ADCT (R,R)	19	7.33
35*	Face	200	DPCM (R,R)	18	6.8	PVQ (L,L)	01	10.94
38	Face	200	PVQ (R,R)	03	1.54	ADCT (L,L)	16	2.52
41*	House with sky	200	ADCT (L,L)	18	1.85	DPCM (R,R)	01	3.88
52	House with sky	200	PVQ (L,L)	04	2.29	DPCM (R,R)	15	3.14
65*	Face	200	DPCM (L,L)	19	10.94	PVQ (R,R)	00	6.8
67*	House with trees	200	DPCM (R,R)	18	15.29	PVQ (L,L)	01	13.5

6.0 RECOMMENDATIONS

The ADCT compression technique was demonstrably better at providing better quality images at higher compression than any of the other three compression techniques. Coupled with its ease of implementation (the ADCT is a fixed transform, known to both transmitter and receiver), the ADCT is a good choice for being included in Group 4 for processing gray scale imagery. None of the other three compression techniques, BPC, PVQ, or DPCM, did as well as ADCT in this subjective study. Secondly, the ADCT is the basis of all lossy coding processes specified by JPEG (JPEG's lossless coding processes are based upon DPCM). This makes the ADCT a prime candidate for coding gray scale and color imagery in Group 3 and Group 4.

An interesting caveat of this study, although not totally unexpected, is the discovery that RMS error is not necessarily a good indication of image quality when comparing output images from *different* compression techniques. Apparently, how the compression techniques process the images, and how well a compression technique can disguise errors ultimately dictates how well a compression technique's output image is judged subjectively. An item for future study might be to investigate at what RMS error value output images from different compression techniques are subjectively judged equal in quality.

REFERENCES

[1]. W. B. Pennebaker, J. L. Mitchell, G. G. Langdon, Jr., and R. B. Arps, "*An Overview of the Basic Principles of the Q-Coder Adaptive Binary Arithmetic Coder,*" IBM J. Res. Develop., 32, 1988

[2]. Mitchell and Pennebaker, "*Software Implementation of the Q-Coder,*" IBM J. Res. Develop., 32, 1988

[3]. Huang, H-M and Woods, J. W., "Predictive Vector Quantization of Images," IEEE Transactions on Communication, V. COM-33, No. 11, November 1985, pp. 1208-1219.

[4]. A. Rosenfeldt and A. Kak, *Digital Picture Processing*, Second Edition, Volume 1, pp. 158-159, Academic Press, Inc.

[5]. A. Rosenfeldt and A. Kak, *Digital Picture Processing*, Second Edition, Volume 1, pp. 121-, Academic Press, Inc.

[6]. A. Rosenfeldt and A. Kak, *Digital Picture Processing*, Second Edition, Volume 1, pp. 155, Academic Press, Inc.

[7]. Bethea, Duran, and Boullion, *Statistical Methods for Engineers and Scientists*, Second Edition, pp. 11-12, Marcel Dekker, Inc.

[8]. Bethea, Duran, and Boullion, *Statistical Methods for Engineers and Scientists*, Second Edition, pp. 55-, Marcel Dekker, Inc.

[9]. Delta Information Systems, Inc., "*Computer Simulation of Gray Scale Compression Techniques for Group 4 Facsimile,*" May 1986, DCA100-83-C-0047, Task 84-002, National Communications System.

[10]. Delta Information Systems, Inc., "*Investigation of Vector Quantization for the Coding of Gray Scale Images for Group 4 Facsimile,*" August 1988, DCA100-87-C-0078, Task 87-003, National Communications System

[11]. Delta Information Systems, Inc., "*Investigation of the Adaptive Discrete Cosine Transform Technique for Group 4 Facsimile,*" August 1989, DCA100-87-C-0078, Task 88-007, National Communications System